Final Project

BIG DATA (CE – 610)

Auto trading the S&P500 index using machine learning models

Author: Amin Khoshkenar

1- Introduction

The stock market enables companies to raise capital for growth while providing an avenue for investors to generate wealth. It serves a crucial role in the financial ecosystem, allocating capital efficiently through price discovery and liquidity. However, markets exhibit complex dynamics with secular trends, cycles, noise, jumps, volatility clustering and periods of irrational exuberance and panic. Humans struggle optimizing decisions in the face of uncertainties and emotions. This makes a strong case for machine learning based algorithmic trading. Algorithmic trading automates analyzing data, strategy modeling, execution for improved speed, consistency and returns at lower costs. Machine learning examines vast datasets to uncover complex nonlinear interrelationships impossible via manual techniques. Models can assimilate context-aware information, event-driven news, macroeconomic policies, corporate actions for robust position sizing. High-frequency techniques enable precise entries and exits. Additionally, algorithms remain disciplined by optimizing quantitatively rather than guessing qualitatively. However, algo-trading suffers drawbacks. Strategies optimize back tested historical data, facing model risk. Programming errors can trigger flash crashes through vicious self-reinforcing feedback loops. Over-fitting causes failure to new regimes while incorrectly attributed causations reduce robustness. Regulations require real-time paper trading and auditing before deploying live strategies. Most importantly, extreme events like policy changes, frauds, wars and natural disasters stay unpredictable, causing unanticipated large losses.

2- Project summary

In this project, I examined the S&P 500 stocks' daily prices and volumes between 2017-2023 from Yahoo Finance as this index represents market trajectories being composed of 500 major US public companies. To gauge price momentum and trading activity, I engineered the 20-day Simple Moving Average, 20-day Exponential Moving Average and 3-day Relative Strength Index indicators. The SMA calculates average price equally over 20 days. The EMA prioritizes recent prices more via exponential weighting. The RSI measures the speed of price movements to identify overbought and oversold levels on a scale of 0 to 100. I computed the first differences of prices and volumes to render the time series stationary for analysis since prices follow a random walk.

Correlation analysis between RSI and price changes, and between price change lags and volume changes to determine leading signals was conducted which resulted coefficients range from -1 to 1, quantifying the strength and path of linear relationships. I partitioned the data into training (2017-2018), testing on (2018-2019) and considering (2019-2023) periods as my real data, assuming $1000 initial capital for only long trades

given the secular bull trend. The machine learning model interprets daily trends for entering or exiting positions, accumulating profits and losses accordingly.

I explored a variety of machine learning classification techniques to predict daily price direction including Logistic Regression, Random Forest, Naive Bayes, Linear Discriminant Analysis (LDA) and Support Vector Machines. Logistic Regression models binary outcomes using a logistic function. Random Forest combines multiple decision trees voting on the output class. Naive Bayes applies Bayes theorem assuming strong feature independence. LDA finds a linear combination separating classes optimally. Support Vector Machines determine hyperplanes maximizing class differentiation. Then, I evaluated model performance on unseen test data using precision, sensitivity, accuracy and specificity. Precision measures positive class predictions' exactness. Accuracy conveys overall correctness of labels predicted. LDA achieved the best test performance, followed by Logistic Regression, Random Forest, Support Vector Machines and Naive Bayes in the given order. LDA's superior accuracy arises from its underlying assumption of normally distributed classes - appropriate for stock returns exhibiting a Gaussian random walk.

The cumulative account balance over 2019–2023 demonstrates profitable performance, underscoring the machine learning model's efficacy in internalizing market dynamics conceptually (Figure 1). This serves as a template for automating algorithmic trading systems using Python. However, limitations exist regarding external events. Policy announcements, fraud revelations, geopolitics, natural disasters stay unpredictable but significantly sway prices. Instituting stop-loss limits as percentages instead of absolute dollar values allows better risk control.

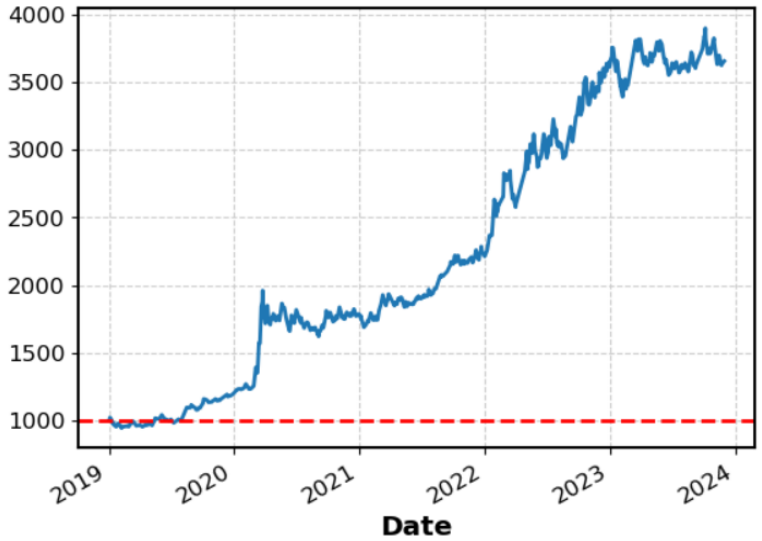

Figure 1. Accumulative balance during real data

Final Project

BIG DATA (CE – 610)

Auto trading the S&P500 index using machine learning models

Author: Amin Khoshkenar

Future enhancements around Bayesian deep learning, Monte Carlo simulation and liquidity flows hold promise. With enough data, computing firepower and financial incentive, machines will optimize decisions better than flawed gut instinct. However, safeguards around model risks, paper trading and regulations persist necessary given algos trigger vicious feedback loops during times of panic by reacting faster than humans.

In conclusion, algorithmic trading promises favorable speed, consistency and returns by programmatically discerning patterns from substantial data - impossible manually. With rigorous validation and risk management, machinic traders will become widespread. But unpredictable regime changes that abruptly invalidate historically optimized strategies remain an ever-present vulnerability in machine learning based approaches.